

Abstract

Processor pipeline controlling techniques are described which take advantage of the variation in critical path lengths of different instructions to achieve increased performance. By examining a processor's instruction set and execution unit implementation's critical timing paths, instructions are classified into speed classes. Based on these speed classes, one pipeline is presented where hold signals are used to dynamically control the pipeline based on the instruction class in execution. An alternative pipeline supporting multiple classes of instructions is presented where the pipeline clocking is dynamically changed as a result of decoded instruction class signals. A single pass synthesis methodology for multi-class execution stage logic is also described. For dynamic class variable pipeline processors, the mix of instructions can have a great effect on processor performance and power utilization since both can vary by the program mix of instruction classes. Application code can be given new degrees of optimization freedom where instruction class and the mix of instructions can be chosen based on performance and power requirements.